



**Web Integration Manual**

**intelliBAS Version 2.7.0.0**

**Release Version 2.7.0.0**

# 1 Table of Contents

<b>1.0</b>	<b>Introduction.....</b>	<b>3</b>
<b>2.0</b>	<b>Prerequisites and Compatibility.....</b>	<b>4</b>
<b>3.0</b>	<b>Integration .....</b>	<b>5</b>
<b>4.0</b>	<b>Functions .....</b>	<b>9</b>
4.1	Functions to be implemented.....	9
4.1.1	OnError Function .....	9
4.1.2	OnResult Function.....	10
4.2	Functions to be called.....	14
4.2.1	SetLanguage Function.....	14
4.2.2	Invoke Function <sup>NEW</sup> .....	14
4.2.3	GetBASResultData Function.....	17
<b>5.0</b>	<b>Plugins and Results.....</b>	<b>18</b>
<b>6.0</b>	<b>Errors and their Description <sup>NEW</sup> .....</b>	<b>19</b>
<b>7.0</b>	<b>Xml Data Format and Samples.....</b>	<b>20</b>
4.3	7.1 Xml Data Format .....	20
4.3.1	7.1.1 BAS Result Xml Data Format.....	20
4.3.2	7.1.2 Twain Setting Xml Data Format <sup>NEW</sup> .....	22
4.3.3	7.1.3 FileUpload Setting Xml Data Format <sup>NEW</sup> .....	23
<b>8.0</b>	<b>Remarks .....</b>	<b>25</b>

## 1.0 Introduction

---

The IntelliBAS is used to acquire biometric data like documents, capture image, scan passport, scan finger print, and read/write card from website. IntelliBAS provides rich interface to acquire such biometric data from any pc integrated with it.

IntelliBAS can be integrate to any website designed on any platform like as JSP, .NET, PHP or HTML. This document describes how to integrate IntelliBAS with the website.

## 2.0 Prerequisites and Compatibility

---

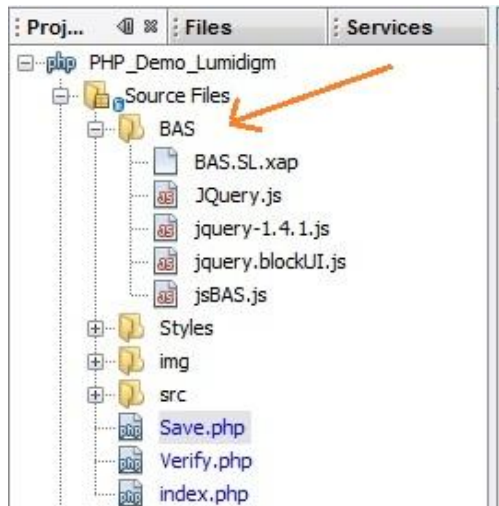
Below you'll find the system requirements for integrating the IntelliBAS with the website. Always install the latest Microsoft updates and service packs.

- Operating Systems:
  - Windows 7
  - Windows XP
- Prerequisites:
  - Microsoft Silver Light 4.0 or higher (on client side)
- Architecture Compatibility:
  - X86.
  - x64 (in 32 bit mode)
- Browser Compatibility:
  - Internet Explorer
  - Mozilla Firefox
  - Google Chrome
  - Opera

## 3.0 Integration

To integrate IntelliBAS with your website, refer the following steps.

- I. Add BAS folder to your website root folder



- II. Add following code in the body element of your web page in which you would like to access BioDevicePlugins.

```
<div id="silverlightControlHost" style="width: 0px; position:absolute; height: 0px; display:block;" >
<object data="data:application/x-silverlight-2," type="application/x-silverlight-2" width="0px"
height="0px" >
    <param name="windowless" value="true"/>
    <param name="background" value="transparent"/>
    <param name="source" value="BAS/BAS.SL.xap" />
    <param name="onError" value="onSilverlightError" />
    <param name="onReceive" value="onSilverlightReceive" />
    <param name="background" value="white" />
    <param name="onLoad" value="pluginLoaded" />
    <param name="minRuntimeVersion" value="4.0.50826.0" />
    <param name="autoUpgrade" value="true" />
    <a href="http://go.microsoft.com/fwlink/?LinkID=149156&v=4.0.50826.0"
    style="text-decoration:none"> </a>
</object>
</div>
```

- III. Add following JavaScript reference in the web page head tag.

```
<script type="text/javascript" src="BAS/jsBAS.js"></script>  
<script type="text/javascript" src="BAS/jquery.js"></script>  
<script type="text/javascript" src="BAS/jquery.blockUI.js"></script>
```

- IV. Call **SetLanguage (<language Name>)** function to set IntelliBAS application language. If “Inherit Language From Server” setting is enabled in the IntelliBAS host, This function will be able to set language to the bio device plugins and host as well. This function call is optional.

Note: Language should be present in IntelliBAS host Application. If does not exist default language (English) will be used.

- V. Call **\_Invoke (<Plugin Instance Name>, < Storage Name>, < Storage Type>, < Is Background Upload>, <Storage Parameter>, < Plugins Parameters>,<PluginSettings>, <Parameter1>, <Parameter2>, <Parameter3>, <Parameter4>, <Parameter5>)** function to invoke bio device plugins from the website this function should be called. Calling this function will invoke bio device plugin linked with “Plugin Instance Name” on the particular machine, from where the invoke call was made.

- VI. Implement following function in JavaScript:

**\_OnError (errType, errMsg)**

When bio device plugins are invoked and result is acquired, it will return result to the website with particular error code.

E.g. if requested plugin instance is not found in IntelliBAS host application errType will be “Error” and errMsg will be “Plugin Instance not found”.

## Upload Data Options:

Here, you have two ways to transfer data to the host website/web server. These both ways are explained below.

### With Storage Plugin

- Storage plugin will allow host website to update rapidly and upload the biometric data in synchronous and asynchronous modes.
- Asynchronous data transfer mode gives good user experience with high speed.
- BAS has defined interface for Storage Plugin in VB.Net, to implement this method host website has to develop StoragePlugin which implements this interface in .Net.
- Host may develop multiple storage plugins for each Bio device plugin they would like to integrate with the website; they can also combine identical data structure bio device plugin in to one StoragePlugin or can develop one storage plugin to support all the Bio Device plugin.
- On successful data acquisition IntelliBAS will call StoragePlugin linked with the particular bio device plugin.
- “Storage Plugin Development Document.docx” document describes how to develop and deploy Storage Plugin.
- If host website is using asynchronous data transfer mode then IntelliBAS will store acquired data on temporary storage location and return file path, and upload it in background during other execution.
- It will return storage path/URL from storage plugin if host would like to update website without refreshing page or particular control.
- This method can be used if host website already has a web service or would like to develop web service.

### Bypass Storage Plugin

- With this approach IntelliBAS returns XML file which contains the acquired biometric data.
- If no Storage Plugin is linked to a BAS Plugin, BAS Host will automatically return acquired data in XML stream based on a predefined XML format (BASResult.xml) this XML data format is explain in section.
- On successful acquisition, IntelliBAS will return XML path. BAS will return “base64String” string as a result xml.
- Only one result link will be received from the IntelliBAS, website has to call IntelliBAS method to get acquired data XML stream.
- Host website has to Call **\_GetBASResultData(pXmlPath)** function to get IntelliBAS result data(XML Stream), **\_BASResultData(\_pResultXmlString)** function will call by IntelliBAS implemented by host website to pass the acquired result, from which website can upload acquired result on the web server. This extra function call is not required in case if Storage Plugin is used.

- Host application may use following approach to upload result XML string to web server:
  - Call web service function from Java Script
  - Call code behind's web method from Java Script
  - Use Hidden Form Fields to assign result XML string and use that variable to upload the XML
  - Develop SilverLight control to invoke web service and call SilverLight function from Java Script
- Host website has to write logic to update website after uploading result XML to the web server or refresh web page or particular control to update the website.
- This approach can be implemented if host website doesn't want to implement new web service nor do not wish to update the existing web service.

Depending upon the choice made, following step is taken. If you would like to integrate with storage plugin then follow the **VII (a)** else go for the **VII (b)**

#### **VII. (a)** Implement following function in JavaScript

`_OnResult(<pResultType>, <pPluginInstanceName>, <pStorageId>, <pstorageParameters>, <pResultArray>)`

- Display success acquisition message.
- Process each result based on pPluginInstanceName and resultType to update the website content (image, link, personal data etc.) Only process resultArray if URL /data are return from the StoragePlugin.
- Refresh page if data transfer mode is synchronous, Storage Plugin is not returning url/data and website need to update after acquisition, else simply return from the function.

#### **(b)** Implement following function in JavaScript

`_OnResult (<pResultType>, <pPluginInstanceName>, <pStorageId>, <pstorageParameters>, <pResultArray>)`

- Display success acquisition message.
- Process resultArray to get result xml URL from the intelliBAS.
- Call `_GetBASResultData(pXmlPath)` function to get IntelliBAS result data (XML Stream), from IntelliBAS call back function `_BASResultData(_pResultXmlString)` will be called from where host website can be updated.



## 4.0 Functions

---

There are several functions which need to be call and implement in order to implement IntelliBAS successfully in any website. Those functions are listed and explained below.

### 4.1 Functions to be implemented

There are functions to be implementing in host website to integrate IntelliBAS and receive the biometric data from IntelliBAS.

#### 4.1.1 OnError Function

To handler information, warning, error or notification from IntelliBAS, this function is to be implemented. This function is must to implement.

Prototype:

`_OnError (errType, errMessage)`

Arguments:

- `errType (String):`

Error type will be string and it will describe what kind of error message host website has got.

- `errMessage (String):New`

errMessage will be string and will describe the error or notification from the intelliBAS service. Here based error type, application set its error message. It will contain error code and error description. If host website wants to display custom message errMessage should be parsed.

Error Message format: "Error Code" + ":" + Error Description

Website can be parsed error code by delimiter ":" and based on error code custom message can be displayed. Refer section 6.0 for detailed error code and there description in detail.

Example:

```
function _OnError(_errType, _errMessage)
{
    //Here, nice popup (with relevant icon) can be used as per
    Website theme instead of alert!
    //SUCCESS_LANGUAGE in case of successful language set
    if (_errType == 'INFO') {
```

```

        alert(_errMessage);

        elseif (_errType == 'WARNING') {
            alert(_errMessage);
        }

        elseif (_errType == 'ERROR') {
            alert(_errMessage);
        }

        elseif (_errType == 'SUCCESS_LANGUAGE') {
            alert(_errMessage);
        }

        return false;
    }
}

```

#### 4.1.2 OnResult Function

This function will be called by intelliBAS host when data acquired successfully for requested plugin. From this call back function host website can be updated. This function must be implemented by the host website.

Refer section 5.0 to get detail what would be result type and number of result of each plugin.

Prototype:

```

_OnResult(<pResultType>,< pPluginInstanceName>, < pStorageId>,
<pstorageParameters>,<pResultArray>)

```

Arguments:

- **pResultType<String>**

Value of this parameter will be "SUCCESS\_ACQUIRED"

- **pPluginInstanceName<String>**

Input parameter which passed as a <Plugin Instance Name> during plugin invoked

- **pStorageId<String>**

Input parameter which passed as a <Storage Name >during plugin invoked

- **pStorageParameter<String>**

Input parameter which is passed as a <Storage Parameter > during plugin invoked

- **pResultArray<Array>**
  - List of result return by the intelliBAS plugin, each element contain resultType (Data type of the each acquired data) and URL/String separated by ";".
  - No. of result will be varying from plugin to plugin and data transfer technique used.
  - If host website has developed StoragePlugin and use asynchronous data transfer intelliBAS service always returns URL/String of temporary storage location.
  - If host website implements StoragePlugin synchronous data transfer method it's up to host website to return storage URL/String from the StoragePlugin, its optional and required only if website want to display biometric data instantly after acquisition without refreshing webpage.
  - The purpose to return data and data type both is to handle very complex scenario. By this, website can differentiate URL/Link/Text data by evaluating data type. Data type can be "Image", "Xml", "PlainText" and "Document", and data can be URL or String if plugin return text data.
  - Different plugin result can be differentiating from Plugin Instance Name.

Reference Block to parse the result array:

```
var result;
for (var i = 0; i < pResultArray.length; i++)
{
    result = pResultArray[i].split(";");
    // result[0]= Data type of result(Image,Xml,PlainText,Document)
    // result[1]= Url/String
    //Set image, document link or text data to update the website
}
```

How \_OnResult() function should be implemented with different data transfer mode?

IntelliBAS supports various device plugins.

Following examples show how to implement \_OnResult () function for "Twain" & "FileUpload" device plugins.

- **With StoragePlugin Asynchronous mode and Synchronous mode if StoragePlugin returns URL/Text**

See example how to implement \_OnResult() function with StoragePlugin Asynchronous and Synchronous data transfer mode.

//Note: On adding new Device Plugin rather than "File Upload" & "Twain" it's required to add new elseIf block based on PluginInstanceName

```
function _OnResult(pResultType, pPluginInstanceName, pStorageId,
pStorageParameter, pResultArray)
{
    var result;
    var i = 0;
    //-----FileUpload Plugin-----
    if (pPluginInstanceName == "FileUploadTest") {
        for (i = 0; i < pResultArray.length; i++) {
            result = pResultArray[i].split(";");
            //Write code here to set a link
            //result[0] = datatype and result[1] = url
        }
    }
    //-----Twain Plugin-----
    else if (pPluginInstanceName == "TwainTest") {
        for (i = 0; i < pResultArray.length; i++) {
            result = pResultArray[i].split(";");
            //Write code here to set image
        }
    }
    return false;
}
```

- **With StoragePlugin Synchronous mode if StoragePlugin does not return URL/Text**

See example how to update the website if StoragePlugin Synchronous mode is used and if StoragePlugin is not returning storage path or data for “File Upload” and “Twain” Device plugin.

```
function _OnResult(pResultType, pPluginInstanceName, pStorageId,
pStorageParameter, pResultArray)
{
    //-----FileUpload Plugin-----
    if (pPluginInstanceName == "FileUploadTest") {
        alert("File Uploaded Successfully...");
    }
    //-----Twain Plugin-----
    if (pPluginInstanceName == "TwainTest") {
        alert("Photo Captured Successfully...");
    }
    return false;
}
```

- **Without Storage Plugin**

See example how to implement \_OnResult() function if StoragePlugin is bypass

```
//Note: On adding new Device Plugin rather than "File Upload" & "Twain" it's
required to add new elseIf block based on PluginInstanceName

function _OnResult(pResultType, pPluginInstanceName, pStorageId,
pStorageParameter, pResultArray)
{
    if (pPluginInstanceName == "FileUploadTest") {

        var result;
        for (var i = 0; i < pResultArray.length; i++) {
            result = pResultArray[i].split(";");
            //Call BAS method to get acquired data
            // result[0] contains data type which is "Xml" and
            // result[1] contains XML URL
            _GetBASResultData(result[1]);
        }
    }
    return false;
}
```

For other device plugins, please refer device specific manuals.

## 4.2 Functions to be called

There are functions from IntelliBAS which need to be called for implementing IntelliBAS in any website and receive biometric data.

### 4.2.1 SetLanguage Function

To set the desired language in the IntelliBAS, this function needs to be called.

If Inherit Language from Server setting is enabled in the IntelliBAS, it will accept this language as a new language and will set the corresponding nodes. It should be called before device plugin invoke.

Prototype:

`SetLanguage (<language Name>)`

Arguments:

- **Language Name (String):**  
To set the desired language in IntelliBAS, language name is passed as a string via this argument to the SetLanguage function.

### 4.2.2 Invoke Function <sup>NEW</sup>

Invoke function invokes the bio device plugin on the machine configured with IntelliBAS. There are number of arguments in Invoke function. Call this function whenever bio metric data needs to be acquired from IntelliBAS service. Change username and password.

Prototype:

`_Invoke (<Plugin Instance Name>, <Storage Name>, <Storage Type>, <Is Background Upload>, <Storage Parameter>, <Plugins Parameters>, <PluginSettings>, < Parameter1>, < Parameter2>, <Parameter3>, <Parameter4>, <Parameter5>)`

Arguments:

- **PluginInstanceName (String):**  
Name of the plugin instance associates with BioDevicePlugin, set from IntelliBAS application  
  
E.g., to upload the document, select File Upload plugin in IntelliBAS application add one instance with name "FileUploadTest", which should be passed as <Plugin Instance Name> parameter in invoke function.
- **StorageName (String):**  
This attribute is used to uniquely address/save the acquired biometric data.

- **Storage Type (String-Optional):**

This attribute will be used by the storage web service to store the acquired data in correct file location or correct data.

- **IsBackgroundUpload (Boolean):**

True (Asynchronous Process): Acquired data will be stored temporarily on client pc and uploaded by intelliBAS service. The website does not required data before continuing.

False (Synchronous Process): Acquired data will instantly upload to the website. The website is not able to continue without having data.

- **StorageParameters (String-Optional):**

This attribute is used to provide the BioDevice Plugin additional parameters.

Eg. If we create a BioDevice Plugin that adds watermark to the image, we could use these additional parameters-> Sample "Test|Demo". After capturing image watermark will be generated by which this contains 2 strings Test, Demo.

- **PluginParameters (String - Optional):**

This attribute is used to pass plugins parameters to the BioDevicePlugins

E.g. any plugin is developed which used parameters/information from host application then this parameter can be use full.

Host can pass number of parameters to the intelliBAS plugin via this parameter by conceiting any special character like ",", "|" etc.

Example:

if host website want to integrate Twain Plugin in their website, and < Plugin Instance Name> is "Twain\_Photo", storage Type is file system and directory name is "Photo", record id is "506612" and background upload is false then \_Invoke() definition will become,

Assume that host website is not using any credential to store the data,

```
_Invoke ('Twain_Photo','506612' , 'Photo',false,"", "", "", "", "", "", "")
```

Assume that host website uses credential to upload the data and credential is user name, password and customer id. This credential can be passed as a storage parameter concatenated by "|".

StorageParameter = "UserName|Password|CustomerId"

If UserName = TestUser,

Password = TestUser,

CustomerId =109

`_Invoke('Twain_Photo','506612', 'Photo',false,'TestUser|TestUser|109',' ',' ',' ',' ',' ')`

<Storage Parameter> can be encrypted from host website and decrypt in StoragePlugin before use it for the security purpose.

- **PluginSettings (String- Optional):** <sup>New</sup>

This attribute is used to pass plugin settings as an XML string to the BioDevice plugins.

This attribute can also be optionally passed as blank, in which case, the IntelliBAS settings file will be used by default.

E.g. Twain settings xml and FileUpload setting xml. Refer section [7.1.2 Twain Setting Xml Data Format](#) and [7.1.3 FileUpload Setting Xml Data Format](#)

- **Parameter1 (String):** <sup>New</sup>

This attribute is used to pass the UserName to the web application for invoking the plugin.

This feature provides access rights management. By providing credentials only authorized users can invoke the plugin. For invoking the plugin the user's credentials (UserName and Password) should match. It is used for user authentication.

- If the credentials match then, the application can invoke the plugin.
- If the credentials fail to match, a message box with the message "Invalid Credentials" will be displayed.

Alternatively, if the 'User Authentication' feature of the IntelliBAS application is enabled, then this attribute can also be passed "blank" by the website.

- When this attribute is passed as blank then, the user gets a pop-up window the first time plugin is invoked; prompting to enter the 'Username' and 'Password'.
- Once the credentials are validated, they are stored in the local configuration file, from where they are passed to the host application on subsequent invoking of the plugins.

- **Parameter2 (String):**

This attribute is used to pass the "Password" used for user authentication to the web application.

- **Optional Parameters:**

Parameter3, Parameter4 and Parameter5 are the three optional attributes added for future development purposes. They can be passed as empty string.



### 4.2.3 GetBASResultData Function

To get result biometric data from the IntelliBAS for bypass storage plugin data transfer mode, this function needs to be called. Pass acquired URL from `_OnResult ()` function as argument in this function.

And ultimately the host web site will get the acquired biometric data.

Prototype:

`_GetBASResultData (pXmlPath)`

Arguments:

- `pXmlPath (String)`:

The physical path of the result data xml is passed to the function. The xml path should be in proper string format.

## 5.0 Plugins and Results

IntelliBAS support various device plugins. It can invoke and transfer the data in various ways. All the cases of “FileUpload” & “Twain” device plugins are describes below.

Configuration Parameters			Output Parameters	
Plugin Name	Storage Plugin	Synchronous	Data Type	No Of result
Twain	Yes	Yes	Image	1
Twain	Yes	No	Image	1
Twain	No	NA	Xml	1
FileUplaod	Yes	Yes	Document	No of files upload
FileUplaod	Yes	No	Document	No of files upload
FileUplaod	No	NA	Xml	1

Following table shows the parameters received from intelliBAS for “FileUpload” & “Twain, and plugin specific details which will helpful to upload acquired biometric data via StoragePlugin/without StoragePlugin to differentiate each plugin and result type.

Plugin Name	Result Count	Name	Mime Type	Data	Data Type	Remarks
Twain	1	StorageId	Image/JPG	ByteArray()	Image	MimeType will be as per extension, e.g. if extension is configured “bmp” then mime type will be “image/bmp”
FileUpload	No of Selected Files	StorageId_1 For first file, for second StorageId_2 and so on.	Document/ doc	ByteArray()	Selected File from plugin(image/ document)	MimeType will be as per extension, e.g. if extension is configured “bmp” then mime type will be “document/doc”

For other device plugins, please refer device specific manuals.

## 6.0 Errors and their Description <sup>NEW</sup>

The following table describes all the possible errors that occur while working with the IntelliBAS software. This can be used for troubleshooting purposes.

Error Number	Description
Error-101:	Shows when there is error in invoking the device
Error-102:	Shows when the operation is cancelled in between
Error -103:	Shows when the plugin is unable to communicate
Error -104:	Shows when the plugin cannot be invoked
Error -105:	Shows when the application is unable to launch
Error -106:	Shows when the device is not ready
Error -111:	Shows when the settings configured are invalid
Error -112:	Shows when some unknown error occurs
Error -113:	Shows when the setting for inheriting the language from the server is disabled
Error -114:	Shows when the plugin is unable to set the language
Error -115:	Shows when the plugin instance is not found
Error -116:	Shows when the uploading operation is failed
Error -117:	Shows when the Username or password entered is invalid
Error -118:	Shows when the License is invalid
Error -119:	Shows when Advanced Bio Acquisition Service is not running.

Error List Table

### Errors:

Error Message will be displayed whenever intelliBAS encounters an error. The error message displayed will be of the form:

**"Error- Error\_Number:Error\_Message"**

e.g. "Error-106:Device Not Ready"

User can enter their customized error message in place of Error\_Message. This will be displayed when the error occurs.

## 7.0 Xml Data Format and Samples

### 7.1 Xml Data Format

From intelliBAS many plugins return XML as an output due to multiple parameters reading from corresponding device. This section describes the xml data format of each xml in detail.

#### 7.1.1 BAS Result Xml Data Format

If no StoragePlugin is linked to a requested intelliBAS plugin, intelliBAS Host will automatically return acquired data in XML stream based on a predefined XML format (BASResult.xml). To encapsulate all the result in one file, XML structure is better also it can be parsing easily.

XML format for the BASResult XML;

```
<?xml version="1.0" encoding="utf-8" ?>
<BASResult>
  <!-- Plugin Name of acquired data-->
  <Plugin></Plugin>
  <!-- Plugin Instance Name pass by website to the BAS-->
  <PluginInstanceName></PluginInstanceName>
  <!-- Results-->
  <Results>
    <!-- Result node will be repeated, depend on how many result is returned from requested
    BAS plugin-->
    <Result>
      <!-- Result file name given by BAS if applicable-->
      <Name></Name>
      <!-- MimeType for the result data-->
      <MimeType></MimeType>
      <!-- File format of acquired data always contain "." as prefix -->
      <Extension></Extension>
      <!--Actual Data of acquired plugin-->
      <!-- base64string in case of binary data and simple text if plugin return pain text as
      a result-->
      <Data></Data>
    </Result>
  </Results>
</BASResult>
```

The acquired data will be embedded in this XML-stream in the <Result> element (inside <Results> root node). The number of <Result> nodes is depending on the number of result objects (Photos) that are returned from Bio Device Plugin.

Eg. Twain will have one result node.

Structure of the BASResult XML file;

- **<BASResult>** it's a Root Element of the BAS Result XML.
- **<Plugin>** Exact name of the plugin.
- **<PluginInstanceName>** plugin instance name passed by the host website. Multiple Plugin Instances could be linked to the same plugin, so based on the plugin instance acquired data could be processed properly!
- **<Results>** it contains list of result nodes returned from the called Bio Device Plugin.
- **<Result>** child node of **<Results>** node and one or multiple of these nodes could exist.
  - o **<Name>** child node of **<Result>** → Result file given by the intelliBAS Host (based on the Storage Name)
  - o **<MimeType>** child node of **<Result>** → Mime type of the content.
  - o **<Extension>** child node of **<Result>** → content format of acquired data, always contain "." as a prefix
  - o **<Data>** child node of **<Result>** → this node will contain plugin Biometric actual data and 3 formats are supported;
    - Binary → 64 based encoded
    - Text
    - XML → 64 based encoded

Example;

BAS Result.XML

```
<?xml version="1.0" encoding="utf-8" ?>
<BASResult>
  <Plugin></Plugin>
  <PluginInstanceName></PluginInstanceName>
  <Results>
    <Result>
      <Name></Name>
      <MimeType></MimeType>
      <Extension></Extension>
      <Data>BAS64STRING(Fingerprint.XML)</Data>
    </Result>
  </Results>
</BASResult>
```

### 7.1.2 Twain Setting Xml Data Format <sup>NEW</sup>

```
<?xml version="1.0" encoding="utf-16?">
<PluginInstanceInfo>
  <InstanceName>TwainTest</InstanceName>
  <PluginName>Twain</PluginName>
  <Version>2.5.0.0</Version>
  <PluginInstanceSettings>
    <SettingInfo>
      <Name>SelectedDevice</Name>
      <Value>IntelliCAP 2.0</Value>
      <Type>0</Type>
    </SettingInfo>
    <SettingInfo>
      <Name>Filter</Name>
      <Value></Value>
      <Type>0</Type>
    </SettingInfo>
    <SettingInfo>
      <Name>GetDeviceList</Name>
      <Value>IntelliCAP 1.8|sCAP42 Aries 2.3|TWAIN Biometrika Driver
1.8</Value>
      <Type>0</Type>
    </SettingInfo>
    <SettingInfo>
      <Name>PostDeviceSelection</Name>
      <Value>1</Value>
      <Type>0</Type>
    </SettingInfo>
    <SettingInfo>
      <Name>FileExtension</Name>
      <Value>png</Value>
      <Type>0</Type>
    </SettingInfo>
  </PluginInstanceSettings>
  <StoragePluginInfo>
    <StoragePluginName>StoragePlugin Demo</StoragePluginName>
    <Version>1.5.0.0</Version>
  </StoragePluginInfo>
</PluginInstanceInfo>
```

- **<InstanceName>** Plugin instance that is to be invoked
- **<PluginName>** Exact name of the plugin
- **<Version>** Plugin version
- **<PluginInstanceSettings>** This node contain plugin setting detail. Passing setting of plugin that mention inside PluginName node

### 7.1.3 FileUpload Setting Xml Data Format <sup>NEW</sup>

```
<?xml version="1.0" encoding="utf-16?">
<PluginInstanceInfo>
  <InstanceName>FileUploadTest</InstanceName>
  <PluginName>FileUpload</PluginName>
  <Version>2.5.0.0</Version>
  <PluginInstanceSettings>
    <SettingInfo>
      <Name>NumberOfFile</Name>
      <Value>1</Value>
      <Type>2</Type>
    </SettingInfo>
    <SettingInfo>
      <Name>MaxFileSize</Name>
      <Value>30</Value>
      <Type>2</Type>
    </SettingInfo>
    <SettingInfo>
      <Name>SizeType</Name>
      <Value>MB</Value>
      <Type>0</Type>
    </SettingInfo>
    <SettingInfo>
      <Name>FileExtensions</Name>
      <Value>docx,png,gif</Value>
      <Type>0</Type>
    </SettingInfo>
    <SettingInfo>
      <Name>DefaultFileExtensions</Name>
      <Value>pdf,doc,docx,jpg,png,gif</Value>
      <Type>0</Type>
    </SettingInfo>
  </PluginInstanceSettings>
</PluginInstanceInfo>
```

- **<InstanceName>** Plugin instance that is to be invoked
- **<PluginName>** Exact name of the plugin.
- **<Version>** Plugin version
- **<PluginInstanceSettings>** This node contain plugin settings detail. To pass setting to a plugin, mention it inside PluginName node

## 7.2 Demo

- Browse demo folder to find attached ASPX/PHP sample source code with different data transfer mode.
- IntelliBAS demo are available in PHP and ASPX, it contain all the BioDevicePlugin intelliBAS offers and various data transfer mode support by the intelliBAS.
- Browse following links in order to have a look how intelliBAS works with websites;
  - o PHP Demo = <http://bas.syx-care.com/>
  - o ASPX Demo = <http://dev.42soft.eu/bas/>
  - o JSP Demo = Source code available!
- Follow the steps to test intelliBAS;
  - o Install the default installer
  - o Configure Temporary storage location
  - o Configure device attached to system for required plugin (Refer user manual for more detail)
  - o Save Settings
  - o Start testing



## 8.0 Remarks

---

1. By default intelliBAS can freeze the website till data acquired successfully, for Bypass Storage plugin once xml URL is return BAS will unfreeze the website. If you would like to freeze your website still data uploading is finished, remove “\$.unblockUI()” line from BrowserCallback () function of jsBAS .js and put in your success/failure event of your JavaScript’s “upload data” function.

To Access any files/Directory in websites from PHP server it’s mandatory that the Directory browsing is enable for that directory In other words, add a <Directory> node with Allow All property in httpd.conf.